PHONETIC DISTANCE BASED ACCENT CLASSIFIER TO IDENTIFY PRONUNCIATION VARIANTS AND OOV WORDS

Akella Amarendra Babu[1]*, Ramadevi Yellasiri[2] and Akepogu Ananda Rao[3]

[1]JNIAS, JNT University Anantapur, Ananthapuramu, Andhra Pradesh, India
[2]CBIT, Osmania University, Hyderabad, Telangana, India
[3]CSE Department, JNT University Anantapur, Ananthapuramu, Andhra Pradesh, India

ABSTRACT

The state-of-the-art Automatic Speech Recognition (ASR) systems lack the ability to identify spoken words if they have non-standard pronunciations. In this paper, we present a new classification algorithm to identify pronunciation variants. It uses Dynamic Phone Warping (DPW) technique to compute the pronunciation-by-pronunciation phonetic distance and a threshold critical distance criterion for the classification. The proposed method consists of two steps; a training step to estimate a critical distance parameter using transcribed data and in the second step, use this critical distance criterion to classify the input utterances into the pronunciation variants and OOV words.

The algorithm is implemented using Java language. The classifier is trained on data sets from TIMIT speech corpus and CMU pronunciation dictionary. The confusion matrix and precision, recall and accuracy performance metrics are used for the performance evaluation. Experimental results show significant performance improvement over the existing classifiers.

KEYWORDS

Dynamic Phone Warping, Critical Phonetic Distance, machine Learning, Pronunciation variants

1. INTRODUCTION

The nature of the speech signal is unique. Firstly, there is a lack of invariance among various phonemes due to co-articulation effect. The articulators move early in anticipation of the subsequent phonemes. It results in a big difference in acoustic waveform for the same phoneme and a very little difference between some phonemes. Secondly, the length, size and shape of the vocal tract differ from speaker to speaker. It results in generating different formant frequencies for the same phoneme. Therefore, the phonemic descriptions generated for a word will vary and depend on the speaker's accent, mood and the context [1], [2].

ASR systems are trained using transcribed speech corpus and tested with unlabelled test speech. Linguistic experts transcribe the corpus manually, which is time consuming, manpower intensive and extremely expensive. Therefore, it is unviable to transcribe "everyday speech" corpus. The

DOI : 10.5121/sipij.2015.6403

human speech recognition system, on the other hand, has the inbuilt ability to learn from the "everyday speech" without labelling [3], [4]. Mimicking the human speech recognition system will help incorporating this ability in ASR systems.

The process of pattern recognition by humans is obscure [5]; it is inbuilt in humans. However, major advances in high resolution imaging technologies reveal how the human brain learns from the conversation with other humans. On hearing an utterance, the human brain compares it with the words in its memory. It hypothesizes a word which has maximum similarity, checks the context and accepts the same. This process is simple if the pronunciation already exists in the memory. In case, the pronunciation doesn't exist in the memory, it enrols the new pronunciation in its memory and uses the same for future references [6]. This process is termed as unsupervised pronunciation adaptation [7].

The critical step in the above process is to find the word-by-word similarity or in other words, finding the phonetic distance between a pair of words described by their phoneme sequences. In this paper, we present an algorithm called Dynamic Phone Warping (DPW) to find the distance between a pair of words [8]. We developed a critical distance criterion to identify the non-standard pronunciations.

This paper is organized as follows. The next section deals with measuring the distance between a pair of words. The DPW algorithm is explained in detail with the help of examples. The classification algorithm is described in section three. The experimental setup is explained in section four. The fifth section covers the results and discussion. Conclusion and future enhancements are given in the sixth section.

1.1. Relation to Prior Work

In the past, researchers have used phonetic distance measurements for various purposes. Ben Hixon et al. [9] used a modified Needleman-Wunsch dynamic programming algorithm [10] to calculate the phonetic distance between a pair of phoneme sequences. The weighted phonetic substitution matrix is developed based on the CMUDICT pronunciation dictionary and is used to compute the similarity score between a pair of pronunciation variants. The performance of three G2P methods are compared using the above phonetic distance measurements.

Martijn Wieling et al. [11] used Lavenshtein distance algorithm to measure the phonetic distance between dialect variations. Bark scale is used to measure the acoustic distances between vowels to get better human perception.

Michael Pucher et al. [12] investigated the correlation between the word confusion matrix and phonetic distance measures. Three methods are used to measure the distance between a pair of phonemes. Firstly, the standard Lavenshtein distance which uses equal weight for edit operations and the number of edit operations are normalized by word length. Secondly, the overlap of phonemic features is measured with a weighted Jaccard coefficient. Thirdly, perceptual similarities as weights for substitution costs. These distance measurements are used for developing grammars and dialogs.

In this paper, we propose to use the phonetic distance measurements for the classification of the input utterances into pronunciation variants and OOV words.

2. DYNAMIC PHONE WARPING (DPW)

This section introduces the concept of phonetic distance and explains the algorithm for computing the phonetic distance between a pair of words. The above algorithm is illustrated with the help of an example.

The Standard English language has 39 phonemes. The phonetic sound is generated by a set of articulators. When a human being speaks a word, the articulators change their positions temporally to generate a sequence of phonetic sounds. The articulators are the vocal cords, pharyngeal cavity, velum, tongue, teeth, mouth, nostrils, etc. The articulators and the positions they assume while generating a phoneme are called features corresponding to that phoneme.

2.1. Phonetic Distance

Phonetic distance is the distance between a pair of phoneme sequences. It is the cost of edit operations required to convert one phoneme sequence into another.

2.2. Edit Operations

There are three edit operations and there is a cost attached to each of these operations. The operations are substitution, insertion and deletion.

2.2.1. Substitution Operation Cost

The substitution cost is the cost of exchanging one phoneme with the other while converting one phoneme sequence into the other. The phonetic distance between a pair of phonemes can be measured by using the feature sets of two phonemes [13]. Assuming phoneme Pa is generated using a set of m features, Fa, phoneme Pb is generated using a set of n features Fb, the Jaccard Coefficient of similarity between the two phonemes Pa and Pb is given by

$$JC\ (Pa,\ Pb) = k\ *\ (Fa\ \mathrm{I}\ Fb)\ /\ (Fa\ \mathrm{U}\ Fb) \qquad (1)$$

Where k is a constant calibrated for the best results [9], [11], and [12].

The distortion or the phonetic distance between the pair of phonemes is one minus the above value of Jaccard Coefficient. The phonetic distances for 1521 pairs of phonemes are estimated. 39 columns and 39 rows cost matrix table is filled with substitution costs. All costs of substitution are added up and the average substitution cost is computed.

2.2.2 Insert / Delete Operation Cost (Indel)

Insertion operation (Delete operation) is inserting (Deleting) a phoneme in the phoneme sequence while converting the same into another. The cost of insertion (deletion) of a phoneme is computed as half of the average substitution cost. It is called an Indel.

2.3. DPW Algorithm

The DPW algorithm uses dynamic programming for global alignment. Needleman-Wunsch algorithm is modified to suit the usage of the algorithm in automatic speech recognition

applications. It estimates the normalised distance between a pair of phoneme sequences SeqA and SeqB with m and n phonemes respectively. The DPW algorithm is given below.

Algorithm 1: DPW Algorithm

Input: Two phoneme sequences (SeqA and SeqB)
Output: Normalized phonetic distance between SeqA & SeqB

Begin

Step 1: Initialization

Declare a matrix with m rows and n columns.
Initialize the first row and the first column.

for i=1 to m

$$M(i,1) = i * Indel \quad (2)$$

for $j = 1$ *to n*

$$M(1, j) = j * Indel \quad (3)$$

Step 2: Fill up Remaining Entries of the Matrix

Using the following formula:

$$M(i, j) = min\left((Mi-1, j-1) + C(\varphi i, \varphi j), (Mi-1, j) + C, (Mi, j-1) + C\right) \quad (4)$$

Where C (φi, φj) is the cost of replacing phoneme i with phoneme j.

$$C(\varphi i, \varphi j) = 0 \text{ if } \varphi i \equiv \varphi j \quad (5)$$

$$= Cost \text{ of } replacing\ \varphi i\ with\ \varphi j \quad (6)$$

Distance between the two sequences = Value at the bottom right hand corner entry.

$$D(m, n) = M(m,n) \quad (7)$$

Step 3: Calculate Normalized Phonetic Distance Dn

$$Dn = D(m, n) / max(m, n) \quad (8)$$

End

The value at the bottom right hand corner of the matrix table gives the distance between SeqA and SeqB. This distance is normalized by the length of the longer sequence.

2.4. Illustration of DPW Algorithm

The DPW algorithm is implemented using Java. The calculated average substitution cost is 0.36 and the cost one Indel is 0.18. The DPW algorithm is illustrated with an example. Considering the two accents of the word 'MATURITY', the results of the experiment to measure the phonetic distance between these two phoneme sequences using DPW algorithm are given in Figure 1. As shown in the figure, the absolute phonetic distance between the two phoneme sequences is 0.76

and it is normalized by dividing the same with the length of the longest phoneme sequence which in the above illustration is 9. The normalized distance is 0.084

3. ACCENT CLASSIFIER

Experiments are conducted to find the word-by-word phonetic distance using the vocabulary and their pronunciation variants listed in the CMU pronunciation dictionary [14]. Analysis of the results led to the formulation of two important hypotheses. The first hypothesis is that the phonetic distance between a pair of accents of a word is less than the phonetic distance between a pair of different words. This hypothesis is used to classify the input phoneme sequences into pronunciation variants or OOV words. Another hypothesis is that there is a threshold phonetic distance value which distinguishes a pronunciation variant from an OOV word. Both these hypotheses are tested at 99% confidence interval using z statistic.

In this section, we first define the critical phonetic distance concept and the associated parameters which are used to fine tune this critical distance estimation. Then we describe a new algorithm to estimate the critical distance. We call this algorithm as Critical Distance Estimation (CDE) algorithm. It uses the DPW algorithm explained in the previous section.

Alignment:

SeqA = M AH0 - CH UH1 R AH0 T IY0

SeqB = M AH0 T Y UH1 R IH0 T IY0

Absolute Phonetic Distance = 0.76

Maximum Phoneme sequence length = 9

Normalized Phonetic Distance = 0.084

DPW Matrix

		M	AH0	T	Y	UH1	R	IH0	T	IY0
	0.00	0.18	0.36	0.54	0.72	0.90	1.08	1.26	1.44	1.62
M	0.18	**0.00**	0.18	0.36	0.54	0.72	0.90	1.08	1.26	1.44
AH0	0.36	0.18	**0.00**	**0.18**	0.36	0.54	0.72	0.90	1.08	1.26
CH	0.54	0.36	0.18	0.36	**0.54**	0.72	0.90	1.08	1.26	1.44
UH1	0.72	0.54	0.36	0.40	0.58	**0.54**	0.72	0.90	1.08	1.26
R	0.90	0.72	0.54	0.58	0.76	0.72	**0.54**	0.72	0.90	1.08
AH0	1.08	0.90	0.72	0.76	0.79	0.90	0.72	**0.76**	0.94	1.12
T	1.26	1.08	0.90	0.72	0.90	1.08	0.90	0.94	**0.76**	0.94
IY0	1.44	1.26	1.08	0.90	0.94	1.12	1.08	1.12	0.94	**0.76**

Figure 1. Illustration of DPW algorithm

The methodology consists of two steps – training step and the testing step. In the training step, the critical distance is learnt using transcribed data. In the testing step, the critical distance is used to determine whether the occurrences belong to the same word. The experiments are carried

out on the CMU pronunciation dictionary and TIMIT database. Precision, recall and accuracy performance metrics are used to evaluate the classifier.

3.1. Definitions

The parameters used in the estimation of critical distance are defined below.

3.1.1. Critical Distance ($D_{CRITICAL}$)

The parameter $D_{CRITICAL}$ is defined as the threshold value of the phonetic distance which is used to distinguish between a pronunciation variant and an OOV word. If the phonetic distance between a pair of phoneme sequences is less than or equal to $D_{CRITICAL}$, then the pair is pronunciation variants of the same word. Otherwise, two phoneme sequences are pronunciations of two different words.

3.1.2. Parameter γ

The parameter γ is used in the estimation of $D_{CRITICAL}$ using the formula

$$D_{CRITICAL} = \text{Indel} * \gamma \quad (9)$$

where the constant γ is a weight which is used to vary the value of '$D_{CRITICAL}$' such that $0 \leq \gamma \leq 1.0$ in steps of 0.5.

3.1.3. Parameter δ

The phoneme sequences of two accents vary mainly in vowel positions and the phoneme sequences of two different words vary both in the vowel and in consonant positions. The parameter δ is used to vary the value of the substitution cost of exchanging two vowel phonemes in relation to the substitution of exchanging two consonants (or exchange between a vowel and a consonant). The cost of exchanging two vowels Cv (φi, φj) is calculated using the formula

$$Cv(\varphi i, \varphi j) = \delta * Cc(\varphi i, \varphi j) \quad (10)$$

Where Cc (φi, φj) is the substitution cost of exchanging a consonant phoneme with another consonant or a vowel and the parameter δ is varied such that $0.6 < \delta < 1$.

3.2. Performance Evaluation of the Classifier

The output of the accent classifier is captured in the form of a two-by-two matrix called the confusion matrix as shown in Figure 2. The outcome of each instance of the classifier can be counted into one of the four possible categories. The instance which is a pronunciation variant and it is classified as a pronunciation variant, is counted as true positive (TP). The instance which is a pronunciation variant, but is classified as OOV word is false negative (FN). The instance is an OOV word and is classified as OOV word is counted as true negative (TN) and the instance which is an OOV word, but is classified as a pronunciation variant, is counted as false positive (FP) [16].

Data sets are prepared to include instances of both pronunciation variants and pronunciations of different words. The parameters γ and δ are varied and the confusion matrices for pronunciation variant class are prepared. Precision, recall and accuracy metrics are calculated.

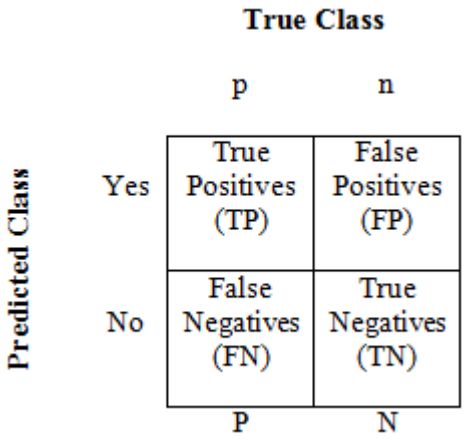


Figure 2. Confusion matrix and the performance metrics

The performance of the classifier is measured using precision, recall and accuracy metrics as under:

$$\text{Precision} = TP / (TP + FP) \quad (11)$$

$$\text{Accuracy} = (TP + TN)/ (P+N) \quad (12)$$

$$\text{Recall (Sensitivity)} = TP / P \quad (13)$$

The precision metric gives better performance evaluation of the classifier than accuracy metric. The accuracy metric combines the true positives and true negatives and maximizes the performance (12). The true negatives, though contribute to the overall ability of classifier, will not contribute towards its ability to classify the positives [17].

3.3. Critical Distance Estimation (CDE) Algorithm

The value of $D_{CRITICAL}$ is estimated empirically using the CDE algorithm.

Algorithm 2: CDE Algorithm

Input: List the base-form pronunciations in input file1 (SeqA) and list the pronunciation variations in input file2 (SeqB)

Output: Value of $D_{CRITICAL}$

Begin

1. Set the value of $D_{CRITICAL}$ with $0 \leq \gamma \leq 1.0$ using the formula (9).
2. Set the value of δ between 1.0 and 0.6 using the formula (10).
3. Select SeqA from file1 and SeqB from file2.

4. Calculate normalized phonetic distance, Dn, between SeqA and SeqB using DPW algorithm.
5. Repeat step 3 and 4 for all SeqA with all SeqB.
6. Count the number of errors in the resolution as per the following criteria:
   a) Set True-Positives = 0; False-Positives = 0;
   b) If SeqA and SeqB are the same word pronunciation variations and Dn <= $D_{CRITICAL,}$ then increment True-Positives
   c) If SeqA and SeqB are different word pronunciations and Dn <= $D_{CRITICAL,}$ then increment False-Positives.
7. Compute precision = True-Positives / (True-Positives + False-Positives)
8. Select the value of Dn as $D_{CRITICAL}$, corresponding to maximum precision.
9. Repeat steps 1 to 8 varying the values of γ, δ and for various data sets.

Select the values of γ and δ at which the $D_{CRITICAL}$ value gives the highest precision.

End

---

4. EXPERIMENTATION

This section details the data sources, cross validation, selection of data sets, and the experimental setup.

4.1. Data sets

Data sets are selected from two data bases; CMU pronunciation dictionary v0.7a (CMUDICT) and TIMIT speech data corpus. The CMU pronunciation dictionary has 134000 orthographic words followed by its phoneme sequences, out of which 8513 words have multiple pronunciations. It consists of isolated words arranged in alphabetical order [14]. The pronunciation phoneme sequences of all words with multiple pronunciations are listed in input file1 as well as in input file2. Each pronunciation variant listed in file1 is compared with pronunciation variants listed in file2.

The TIMIT speech corpus is popularly used in speech recognition research [15]. It consists of connected word speech utterances of 630 speakers, both male and female from eight dialect regions and is 8 KHz bandwidth read speech under the quiet and controlled environment. The training data set consists of utterances from 462 speakers and the test data set consists of utterances from 168 speakers. Each speaker read 10 utterances. The TIMIT data base which is transcribed using 61 labels is mapped to the standard set of 39 phonemes. The transcribed words from the data set are listed in input file1 as well as in input file2. Each pronunciation variant listed in file1 is compared with pronunciation variants listed in file2.

4.2. Five-fold cross validation

A five-fold cross validation consisting of all words in data sets is conducted so that the conclusions are unbiased and demonstrate the principle of repetition in experimentation.

4.3. Grouping and selection of data sets

The CMUDICT is divided into five groups. The words starting with alphabets A to E are put in group 1, the words starting with alphabets F to J are in put in group 2, the words starting with alphabets K to O are put in group 3, the words starting with alphabets P to T are put in group 4 and the words starting with alphabets U to Z are put in group 5. The data sets for estimating the critical distance, are taken from four groups when data sets for testing are taken from the fifth group.

4.4. Experimental set-up

The block diagram of the experimental set up for the estimation of critical distance is given in figure 3.

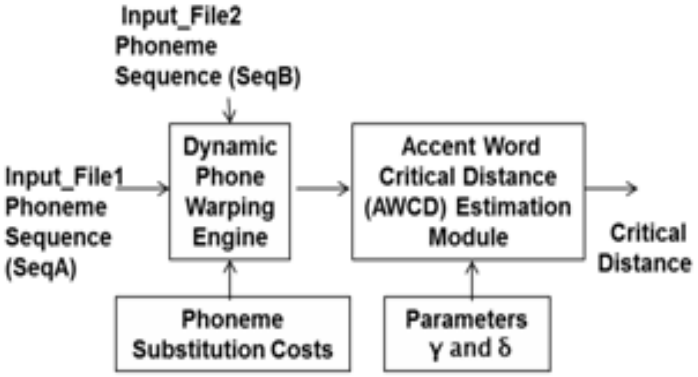


Figure 3. Experimental set up for the estimation of $\mathbf{D_{CRITICAL}}$

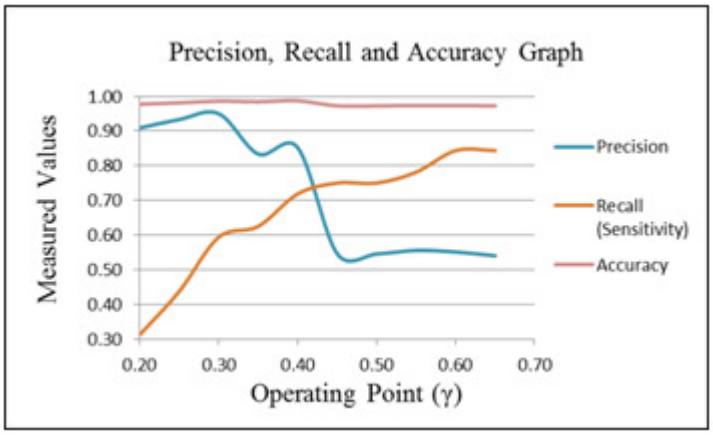
The DPW engine takes phoneme sequences from input_file1 and input_file2 and calculates the normalized phonetic distance between two phoneme sequences. The AWCD estimation module estimates the critical distance by varying the values of γ and δ parameters.

5. RESULTS AND DISCUSSION

Thorough experimentation has been carried out for the estimation of the critical distance using CMUDICT and TIMIT speech corpus. The results of one of the experiments using data sets from TIMIT data corpus are shown in table 1. A portion of the results are graphically shown in figure 4.

Table 1. Results of the experiment for estimation critical distance at δ =0.6

γ	Recall	Precision	Accuracy
0.20	0.313	0.91	0.978
0.25	0.438	0.93	0.981
0.30	0.594	0.95	0.986
0.35	0.625	0.83	0.984
0.40	0.719	0.85	0.987
0.45	0.750	0.55	0.973
0.50	0.750	0.55	0.973
0.55	0.781	0.56	0.974
0.60	0.844	0.55	0.974



Graphical view of the results

In table 1, the precision metric is highest (0.95) at γ = 0.3 whereas the accuracy metric is highest at γ = 0.4. The precision metric is best suited for pronunciation adaptation applications as it is the ratio of true pronunciation variants and total pronunciation variants in the confusion matrix [16]. Therefore, γ=0.3 is taken for the estimation of $D_{CRITICAL}$ in the testing of the pronunciation adaptation model of ASR system applications. The performance metric accuracy is relevant in speech recognition applications as its computation involves both the true positives and true negatives [17]. Therefore, γ = 0.4 is taken for speech recognition applications.

The results of five data sets each from CMUDICT and TIMIT are summarized in Table II. As shown in the table, the precision metric is 100% with data sets from CMUDICT whereas it is between 94% and 96% with data sets from TIMIT data sets. The performance degradation is because CMUDICT consists of isolated words whereas data sets from TIMIT consist of continuous read speech. There is a change in operating point of the parameter γ which is 0.35 for CMUDICT to 0.30 for TIMIT data sets. It is because the classification of pronunciation variants

with TIMIT continuous read speech is more difficult, requiring lower value for the operating point.

Table 2. Performance of the Accent classifier

Data set #	Source of data set	Highest Precision Point		Accuracy		Sensitivity at highest Precision	Accuracy at highest Precision
		Operating Point (γ)	Precision Value	Operating Point (γ)	Value		
1	CMUDICT	0.35	1.00	0.35	0.998	0.903	0.998
2	CMUDICT	0.35	1.00	0.50	0.996	0.839	0.995
3	CMUDICT	0.35	1.00	0.35	0.996	0.839	0.996
4	CMUDICT	0.30	1.00	0.40	0.997	0.774	0.994
5	CMUDICT	0.35	0.96	0.50	0.995	0.839	0.995
6	TIMIT	0.30	0.95	0.35	0.987	0.594	0.986
7	TIMIT	0.30	0.94	0.40	0.982	0.469	0.982
8	TIMIT	0.30	0.83	0.40	0.982	0.313	0.977
9	TIMIT	0.30	0.94	0.40	0.985	0.531	0.984
10	TIMIT	0.30	0.95	0.40	0.985	0.591	0.986

As shown in table 2, accuracy metric is 99% with data sets from CMUDICT and there is slight performance degradation when data sets are taken from TIMIT data base. The operating point of the parameter γ varied between 0.5 and 0.35 in both the cases. It may be attributed to the fact that the accuracy metric is calculated based on both true positives and true negatives (12) which covers both the columns of the confusion matrix. As shown in table 2, there is a slight degradation in the accuracy of the classifier at the highest precision point.

Accuracy is 99% with data sets from CMUDICT and there is slight performance degradation when the data sets are taken from TIMIT data base. The operating point of the parameter γ varied between 0.5 and 0.35 in both the cases. It may be attributed to the fact that the accuracy metric is calculated based on both true positives and true negatives (12) which covers both the columns of the confusion matrix. As shown in table 2, there is a slight degradation in the accuracy of the classifier at the highest precision point.

The recall metric indicates the sensitivity which is calculated as the number of true positives identified out of the total number of positives in the data set. It indicates the rate at which the pronunciation variants are identified and adapted. In other words, it indicates the rate of adaptation of new pronunciation variants.

5.1 Comparison with Existing Classifiers

The speech recognition task is essentially a classification problem. The accent classifier proposed in this paper is an innovative model with specific purpose to classify the given input utterances into pronunciation variants of the existing word in the vocabulary or OOV words. The classifiers are evaluated based on certain performance metrics independent of their purpose and the performance of the Accent classifier described in this paper is compared with the

performance of the other classifiers like Hidden Markov Models (HMM) [19], Artificial Neural Networks (ANN), Conditional Radom fields (CRF) and large margin classifiers based acoustic models.

The state-of-the-art phoneme recognizer is based on HMM-ANN paradigm.The best performance accuracy metric of the HMM-ANN acoustic model trained on the TIMIT database is 73.4% [18]. In comparison, the accuracy performance metric of Accent classifier is 99.7% at the precision performance metric of 83%. The Accent classifier has higher classification performance compared to other classifiers used in the speech recognition task.

6. CONCLUSIONS

We proposed a new classification algorithm which identifies the non-standard pronunciations from the input speech utterances. The DPW algorithm computes the word-by-word and pronunciation-by-pronunciation phonetic distance using dynamic programming. A threshold critical distance criterion is developed to segregate pronunciation variants from the OOV words.

The classifier is evaluated using confusion matrix and the performance metrics; precision, accuracy and recall metrics. The data sets are drawn from the TIMIT database and CMUDICT. The classifier performed better than the other classifiers used in the speech recognition task with a performance accuracy of 99.7% at a precision of 83%.

6.1. Future Directions

The Accent classifier can be used in speaker recognition and speech adaption applications. The pronunciation dictionary may be enhanced with non-standard pronunciation using the data driven approaches. The languages or its dialects with sparse transcribed data corpus may use the Accent classifier to build the customized pronunciation dictionary.

ACKNOWLEDGEMENTS

The experiments are conducted in JNTUA ROSE laboratories and we thank all the staff and the research scholars for their invaluable help and suggestions.

REFERENCES

[1] Huang, Acero & Hon, (2001) “Spoken language processing guide to algorithms and system development”. PH.

[2] Dumpala, S.H., Sridaran, K.V., Gangashetty, S.V., Yegnanarayana, B., (2014) “Analysis of laughter and speech-laugh signals using excitation source information”, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Page(s): pp 975 – 979.

[3] Baker, J. M., Li Deng, Sanjeev Khudanpur, Chin-Hui Lee, James Glass Nelson Morgan, (2009) “Historical developments and future directions speech recognition and understanding”, IEEE Signal Processing Magazine, Vol 26, No. 4, pp 78-85.

[4] Eric Fosler-Lussier, Bonnie J. Dorr, Lucian Galescu, Ian Perera, Kristy Hollingshead-Seitz, (2015) “Speech adaptation in extended ambient intelligence environments”, Proceedings of AAAI, Annual Conference 2015.

[5] S. Pinker, (1997), “How mind works”, Pinguin books, New York, NY.

[6] Rietveld CA., et al., (2014).” Common genetic variants associated with cognitive performance identified using the proxy-phenotype method”. In: Proceedings of the national academy of sciences of the United States of America, Vol 112 (4), 13790.

[7] Babu Akella Amarendra, Ramadevi Yellasiri & A. Ananda Rao, (2014), "Unsupervised Adaptation of ASR Systems Using Hybrid HMM and VQ model", Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2014, Hong Kong, pp 169 – 174.
[8] Rabiner, L., Juang, B. & Yegnanarayana B., (2009), "Fundamentals of speech recognition", Prentice Hall, Englewood Cliffs, N.J.
[9] Ben Hixon, Eric Schneider & Susan L. Epstein, (2011) "Phonemic similarity metrics to compare pronunciation methods", INTERSPEECH 2011.
[10] Needleman, Saul B., Wunsch, Christian D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48 (3): pp443–53.
[11] Martijn Wieling, Eliza Margaretha & John Nerbonne, (2001) "Inducing phonetic distances from dialect variation". Computational Linguistics, Netherlands Journal, pp109-118.
[12] Michael Pucher, Andreas Türk1, Jitendra Ajmera & Natalie Fecher, (2007). "Phonetic distance measures for speech recognition vocabulary", In: Proceedings of 3rd Congress of the Alps Adria Acoustics Association 27–28 September 2007, Graz – Austria.
[13] Gopala Krishna Anumanchipalli, Mosur Ravishankar & Raj Reddy, (2007). "Improving pronunciation inference using n-best list, Acoustics and Orthography", In: Proceedings of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, USA.
[14] Weide, R. L, (1998). "The CMU pronouncing dictionary", http://www.speech.cs.cmu.edu/cgi-bin/cmudict, (5/6/2015).
[15] Garofolo, J. S. et al. (1993), " TIMIT acoustic-phonetic continuous speech corpus", Linguistic Data Consortium, Philadelphia.
[16] Tom Fawcett, (2006), "An introduction to ROC analysis", Elsevier B.V., Science Direct, Pattern Recognition Letters 27, pp861–874.
[17] Jesse Davis & Mark Goadrich, (2006), "The Relationship between precision-recall and ROC curves". In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA.
[18] Joel Pinto, B. Yegnanarayana, Hynek Hermansky & Mathew Magimai. Doss, (2007), "Exploiting contextual information for improved phoneme recognition", IDIAP research report.
[19] Ibrahim Patel, Dr. Y. Srinivas Rao,(2010), Speech recognition using HMM with MFCC- an analysis using frequency spectral decomposion technique, Signal & Image Processing - An International Journal(SIPIJ) Vol.1, No.2, pp 101-110.

AUTHORS

Akella Amarendra Babu received B. Tech (ECE) degree from JNU and M. Tech (CSE) degree from IIT Madras, Chennai. He served Indian Army as Lt Colonel and has senior project management experience in corporate IT industry. He has research experience on mega defence projects in DLRL, DRDO and worked as Professor and HOD of CSE department in Engineering Colleges. He has a few research publications in various national and international conferences and journals. His research interests include Speech Processing, Information Security and Telecommunications. He is a Fellow of IETE, member of CSI and IAENG.

Prof Yellasiri Rama Devi received B.E. from Osmania University in 1991 and M. Tech (CSE) degree from JNT University in 1997. She received her Ph.D. degree from Central University, Hyderabad in 2009. She is Professor, Chaitanya Bharathi Institute of Technology, Hyderabad. Her research interests include Speech and Image Processing, Soft Computing, Data Mining, and Bio-Informatics. She is a member for IEEE, ISTE, IETE, IAENG and IE. She has published more than 50 research publications in various national, inter-national conferences and journals.

Prof. Ananda Rao Akepogu received B.Sc. (M.P.C) degree from Silver Jubilee Govt. College, SV University, Andhra Pradesh, India. He received B. Tech. degree in Computer Science & Engineering and M. Tech. degree in A.I & Robotics from University of Hyderabad, India. He received Ph.D. from Indian Institute of Technology, Madras, India. He is Professor of Computer Science & Engineering and Director of IR & P at JNTUA, Anantapur, India. Prof. Ananda Rao published more than hundred research papers in international journals, conferences and authored three books. His main research interest includes speech processing, software engineering and data mining. He received the best teachers' award from Government of Andhra Pradesh, India, in September 2014.